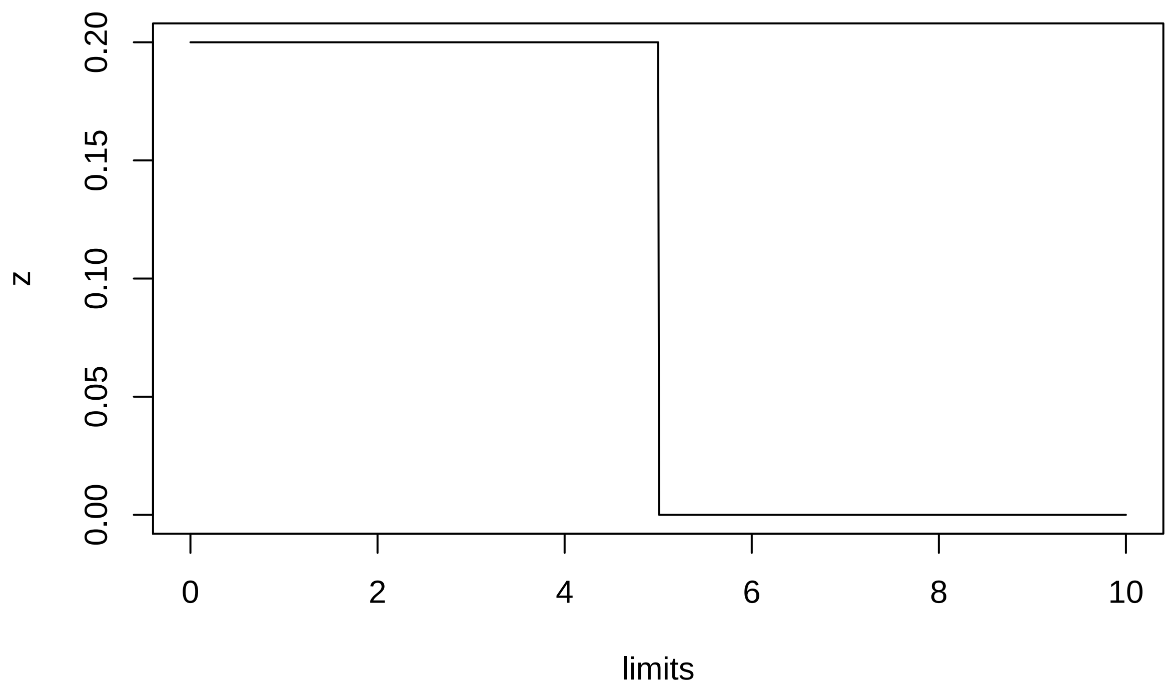
ContinuousProbabilityDistributions.R

Administrator

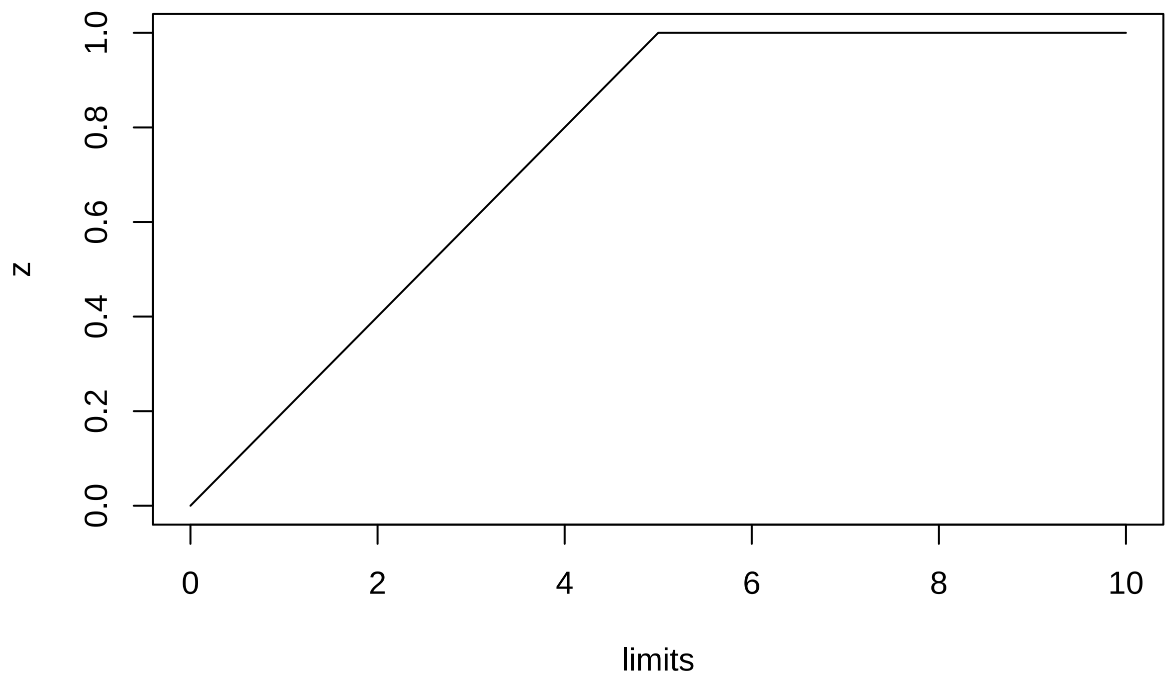
Tue Mar 29 15:00:33 2016

```
# Continuous probability distributions
# 24 March 2016
# NJG
```

```
# uniform
# params specific minimum and maximum

# dunif for density plot
limits <- seq(0,10,by=0.01)
z <-dunif(x=limits,min=0,max=5)
names(z) <- limits
plot(x=limits, y=z,type="l",xlim=c(0,10))
```



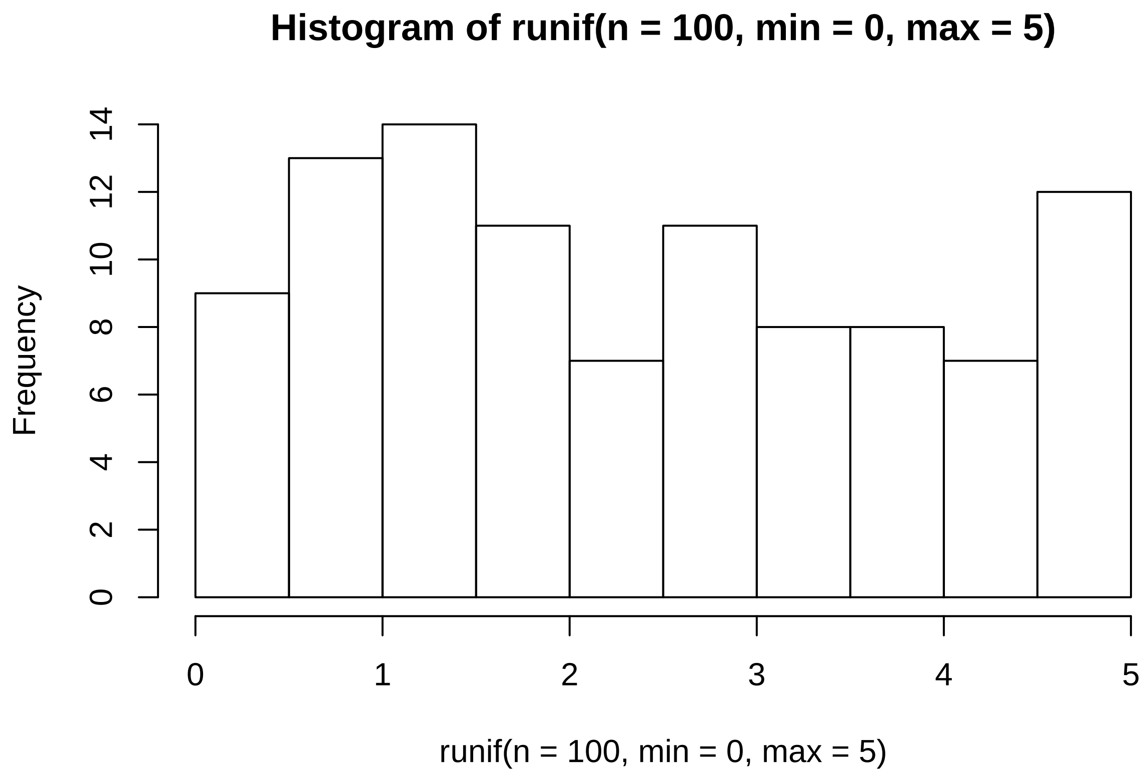

```
#punif for cumulative density (= tail probabilities)
limits <- seq(0,10,by=0.01)
z <-punif(q=limits,min=0,max=5)
names(z) <- limits
plot(x=limits, y=z,type="l",xlim=c(0,10))
```



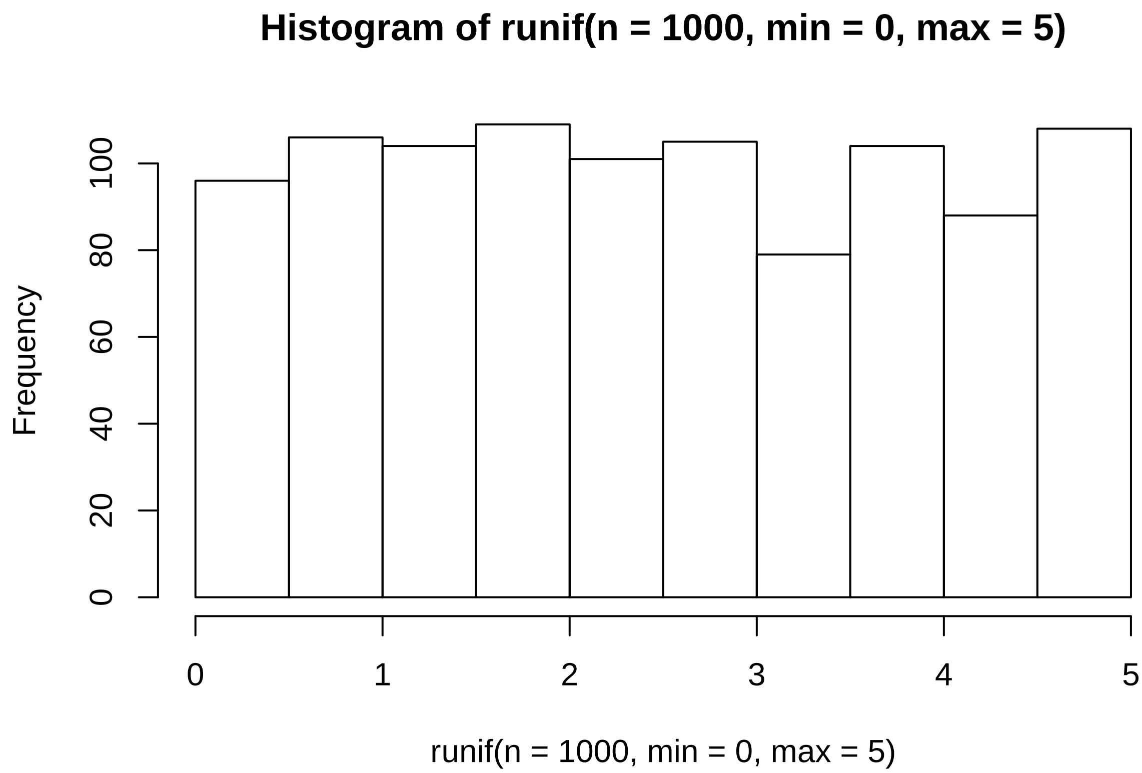

```
#qunif for quantiles
qunif(p=c(0.025,0.975),min=0,max=5)
```

```
## [1] 0.125 4.875
```

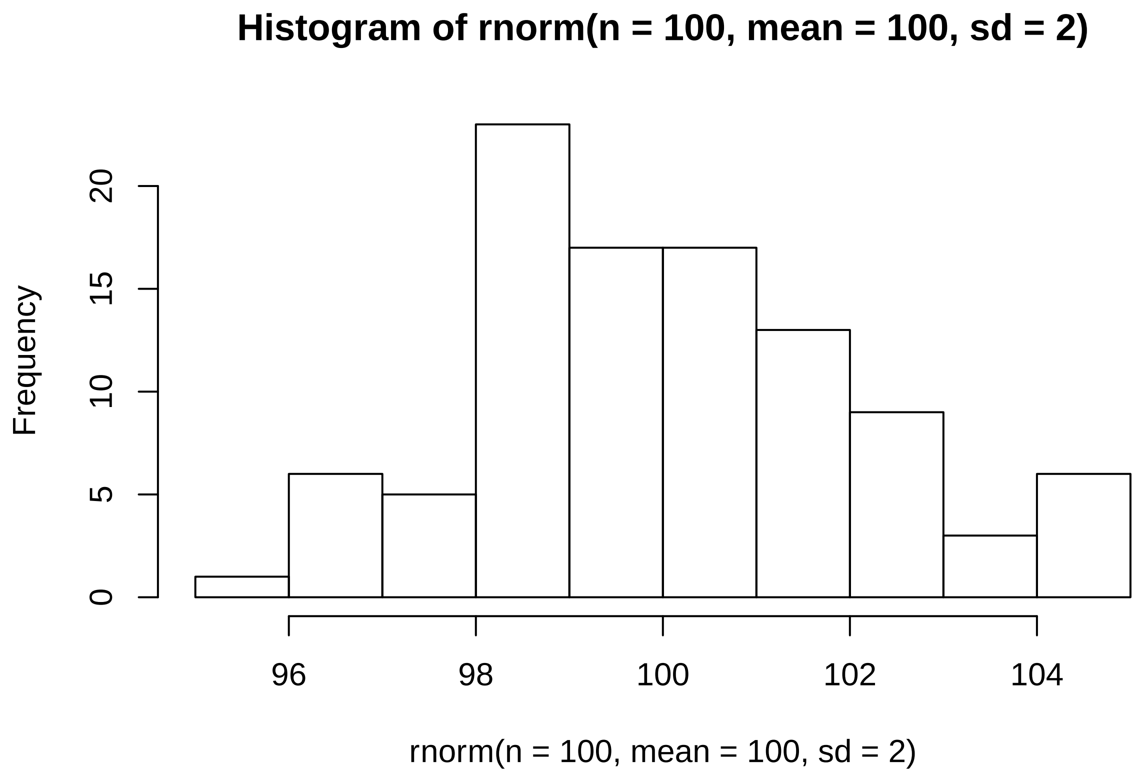
```
#runif for random data
hist(runif(n=100,min=0,max=5))
```



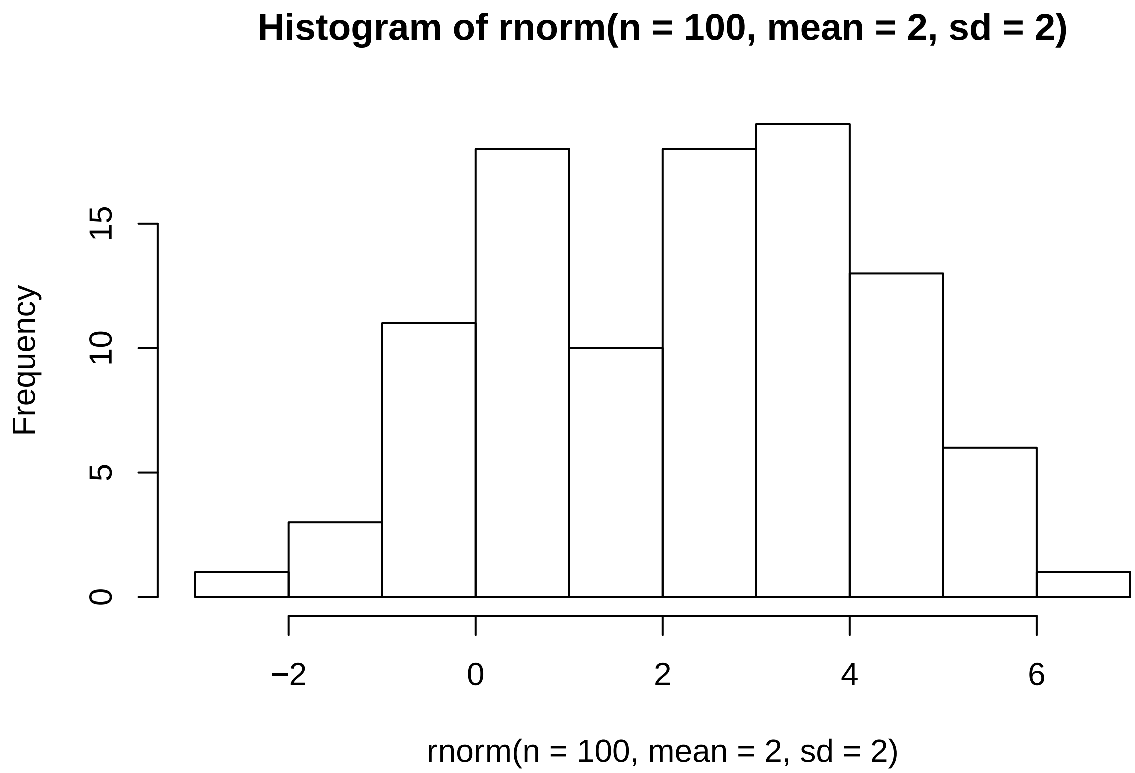

```r
hist(runif(n=1000,min=0,max=5))
```




```
# normal

hist(rnorm(n=100,mean=100,sd=2))
```




```
# problems with uniform when mean is small but zero is not allowed.
hist(rnorm(n=100,mean=2,sd=2))
```




```
MyVec <- rnorm(n=100,mean=2, sd=2)
summary(MyVec)
```

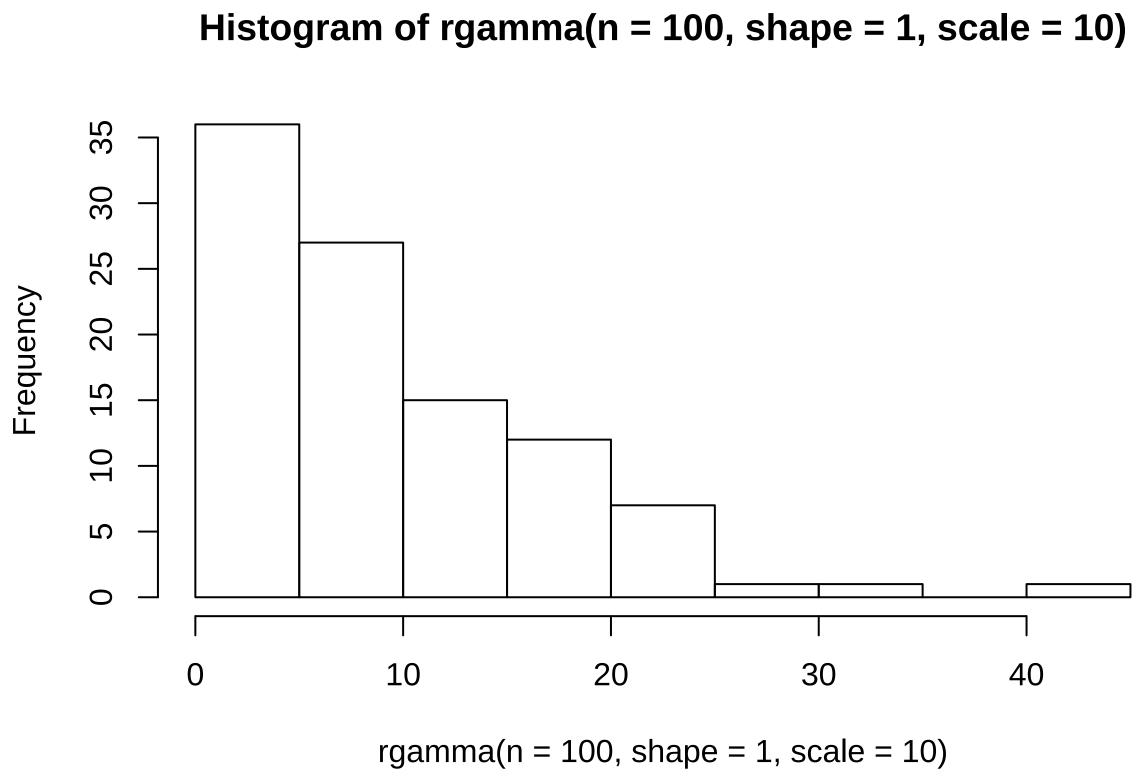
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.8790  0.8296  2.1850  2.3340  4.0100  7.4190
```

```
TossZeroes <- MyVec[MyVec>0]
hist(TossZeroes)
```

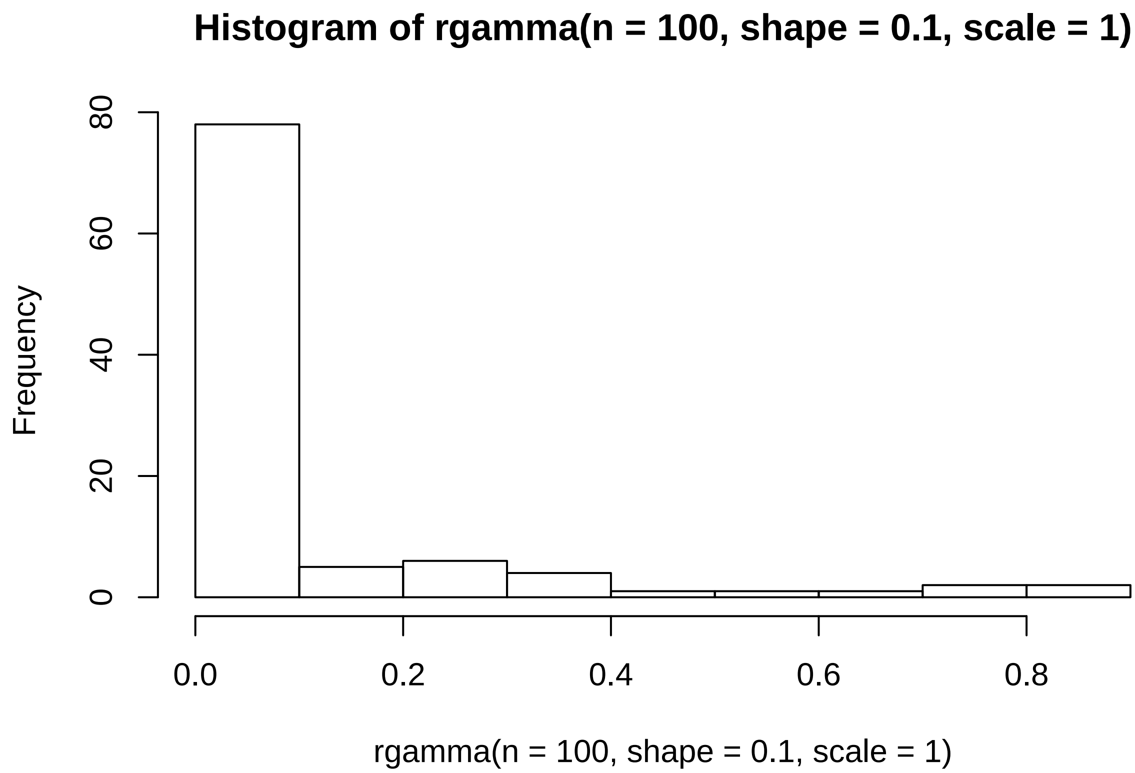

```
summary(TossZeroes)
```

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000603 1.388000 2.625000 2.845000 4.156000 7.419000
```

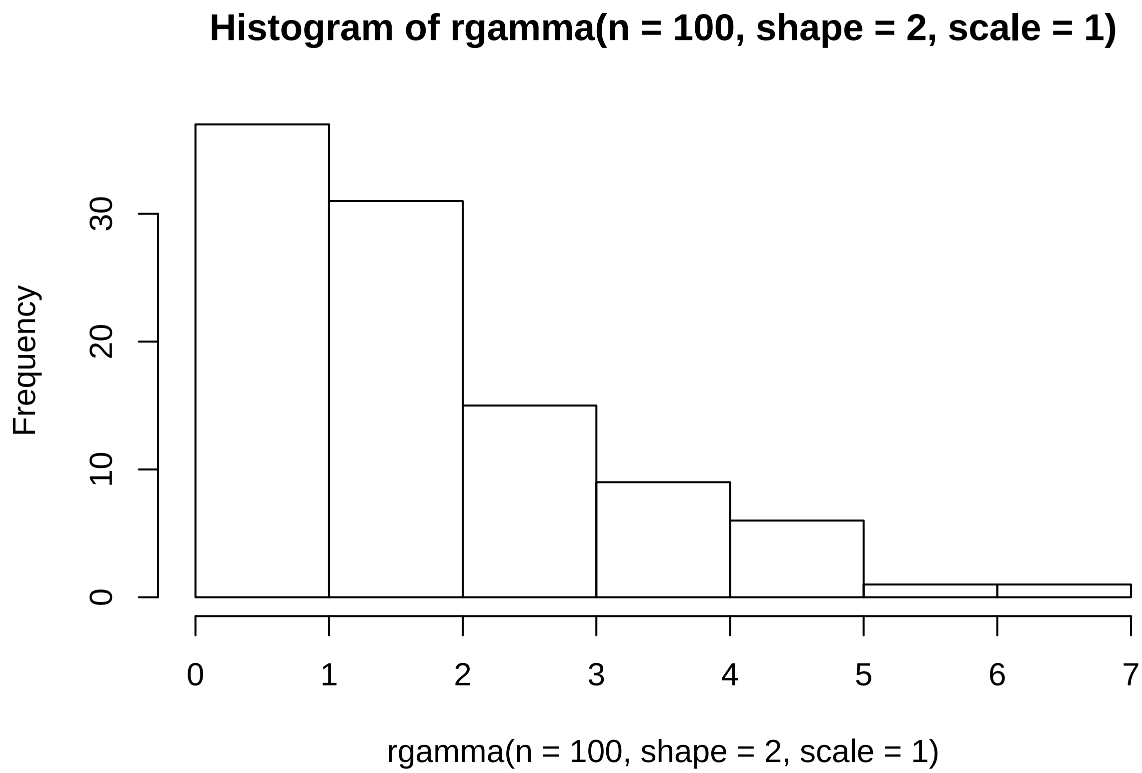
```
# gamma distribution, continuous positive values, but bounded at 0

hist(rgamma(n=100,shape=1,scale=10))
```



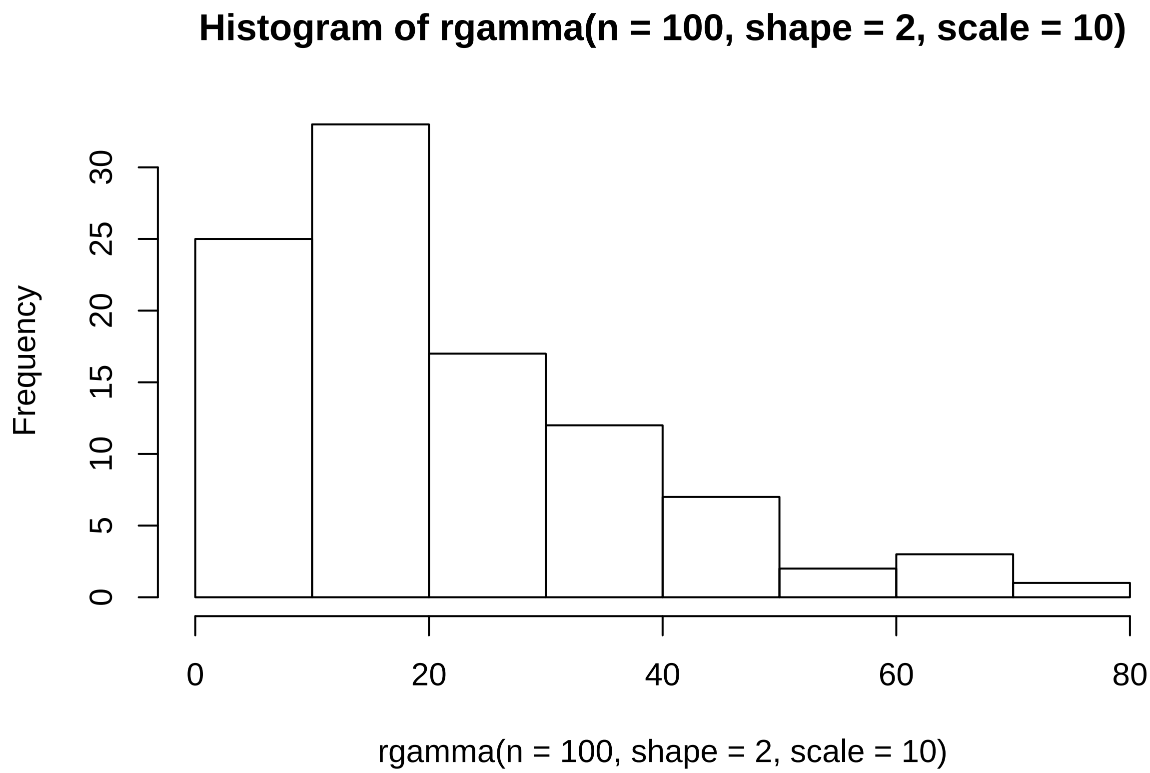

```r
# gamma with shape= 1 is an exponential with scale = mean

# shape <=1 gives a mode near zero; very small shape rounds to zero
hist(rgamma(n=100,shape=0.1,scale=1))
```



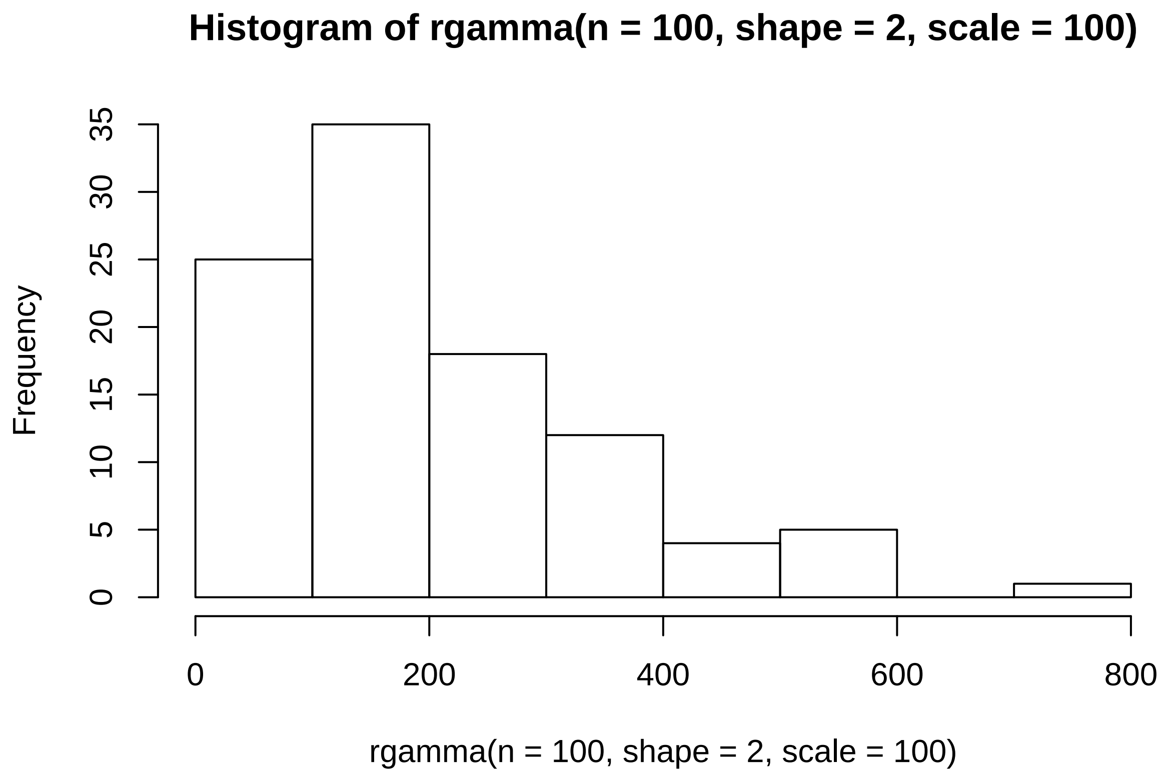

```r
# large shape parameters moves towards a normal
hist(rgamma(n=100,shape=2,scale=1))
```



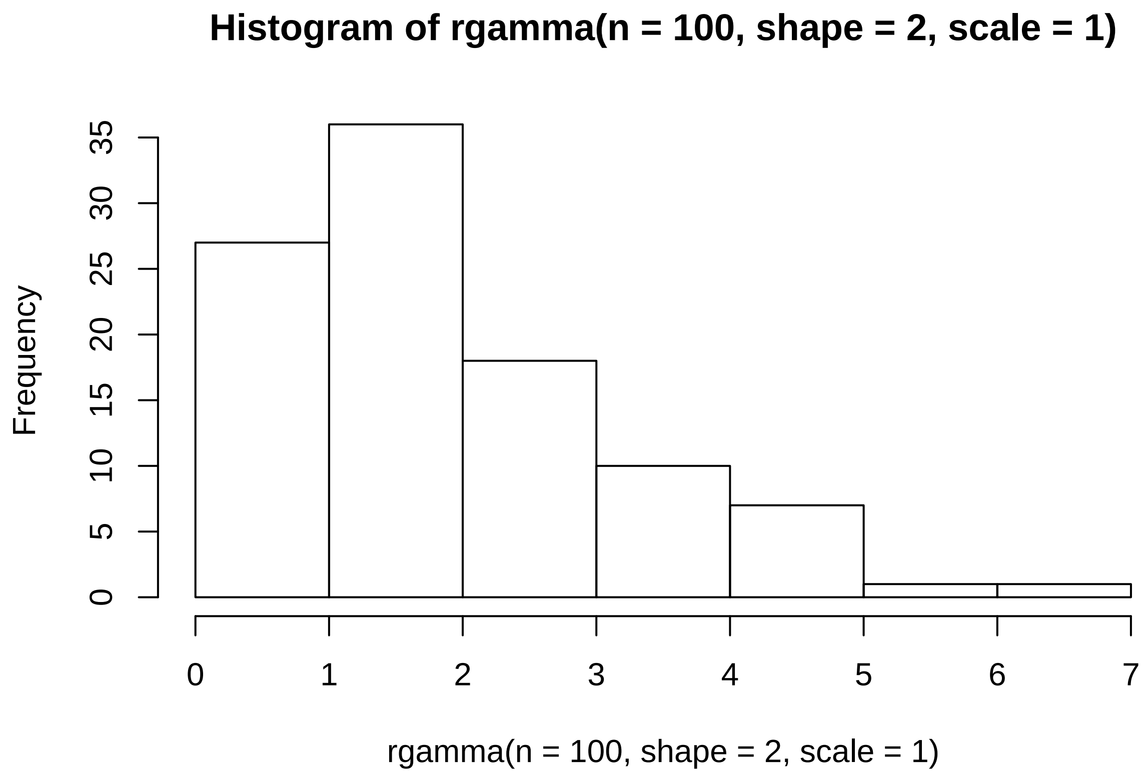

```r
# scale parameter changes mean- and the variance!

hist(rgamma(n=100,shape=2,scale=10))
```



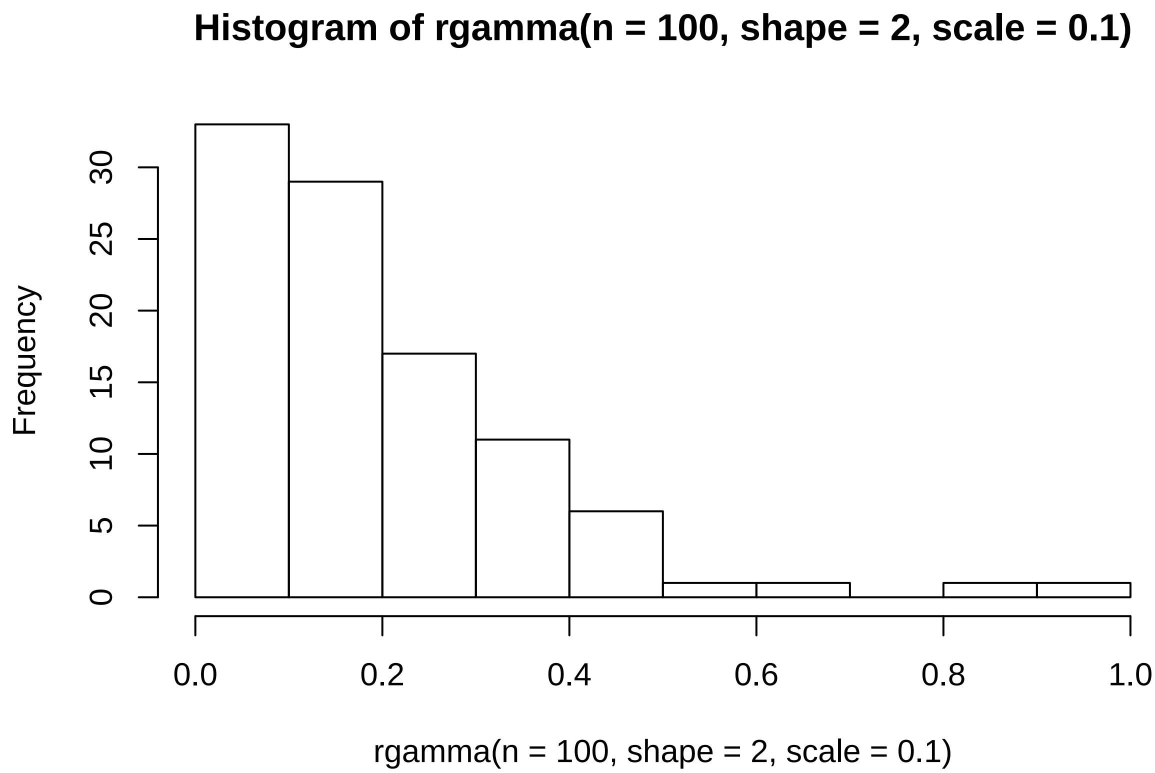

```
hist(rgamma(n=100,shape=2,scale=100))
```




```
hist(rgamma(n=100,shape=2,scale=1))
```

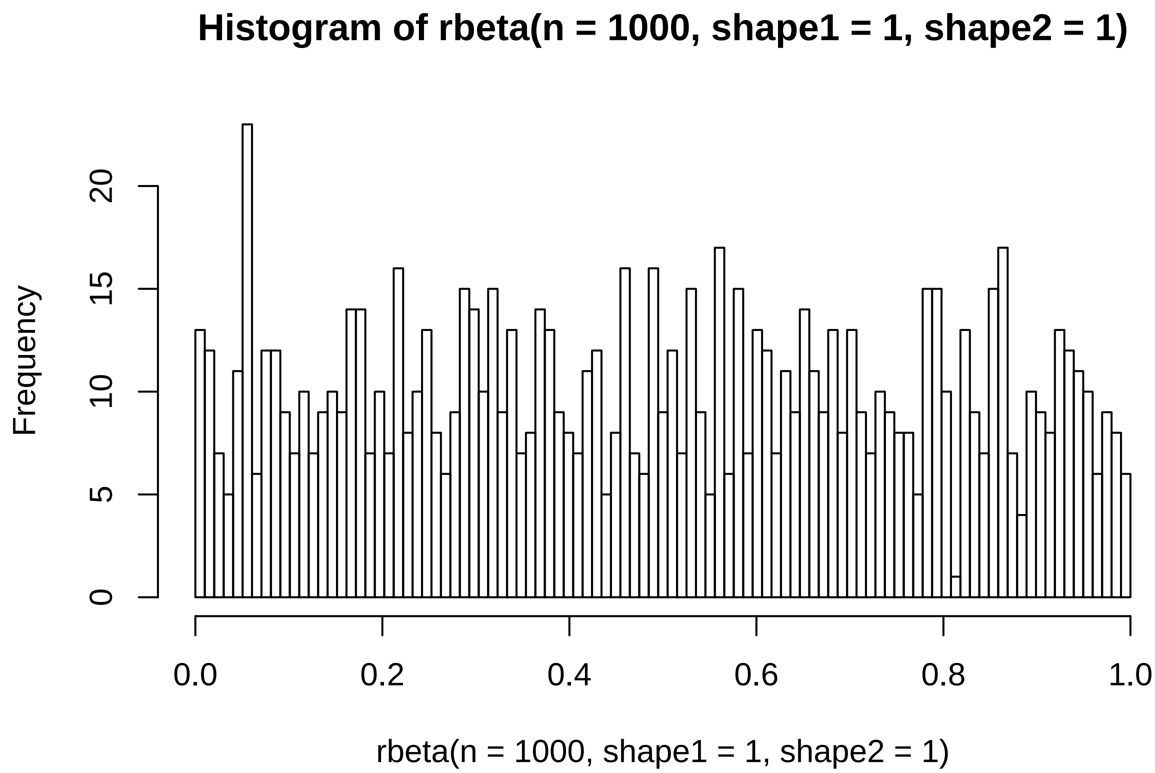



```r
hist(rgamma(n=100,shape=2,scale=0.1))
```

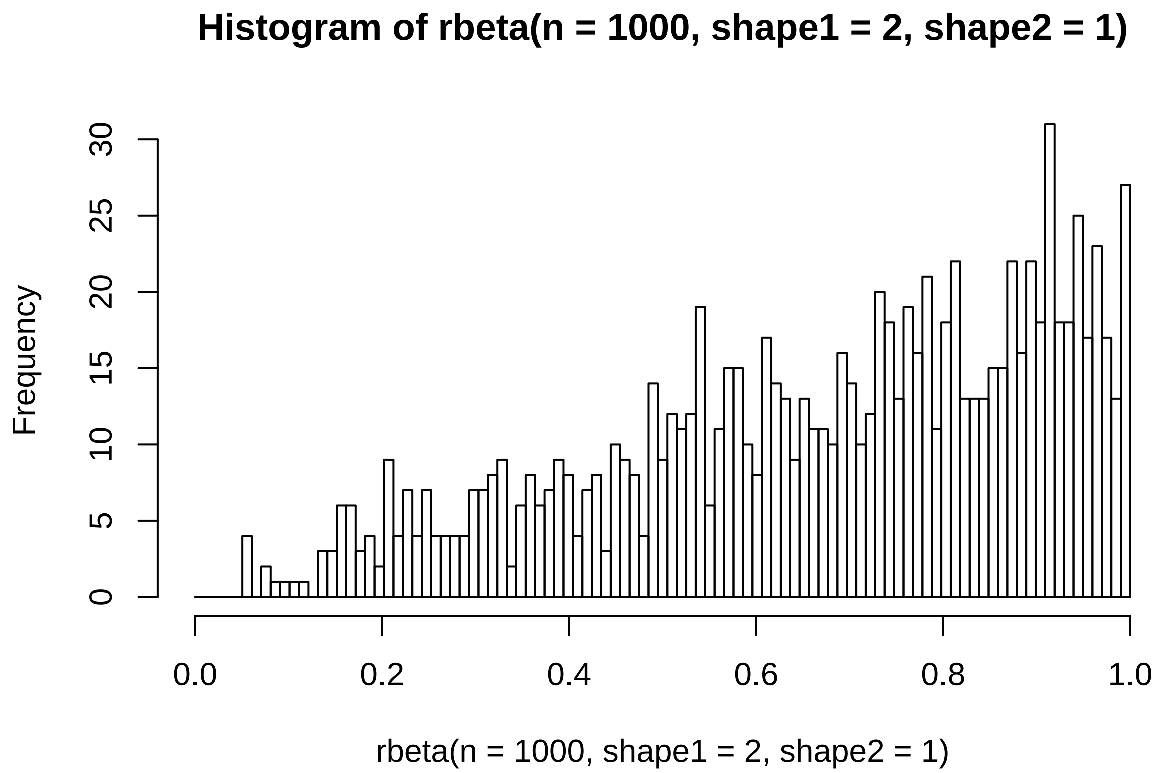



```r
# unlike the normal, the two parameters affect both mean and variance

# mean = shape*scale
# variance= shape*scale^2
```

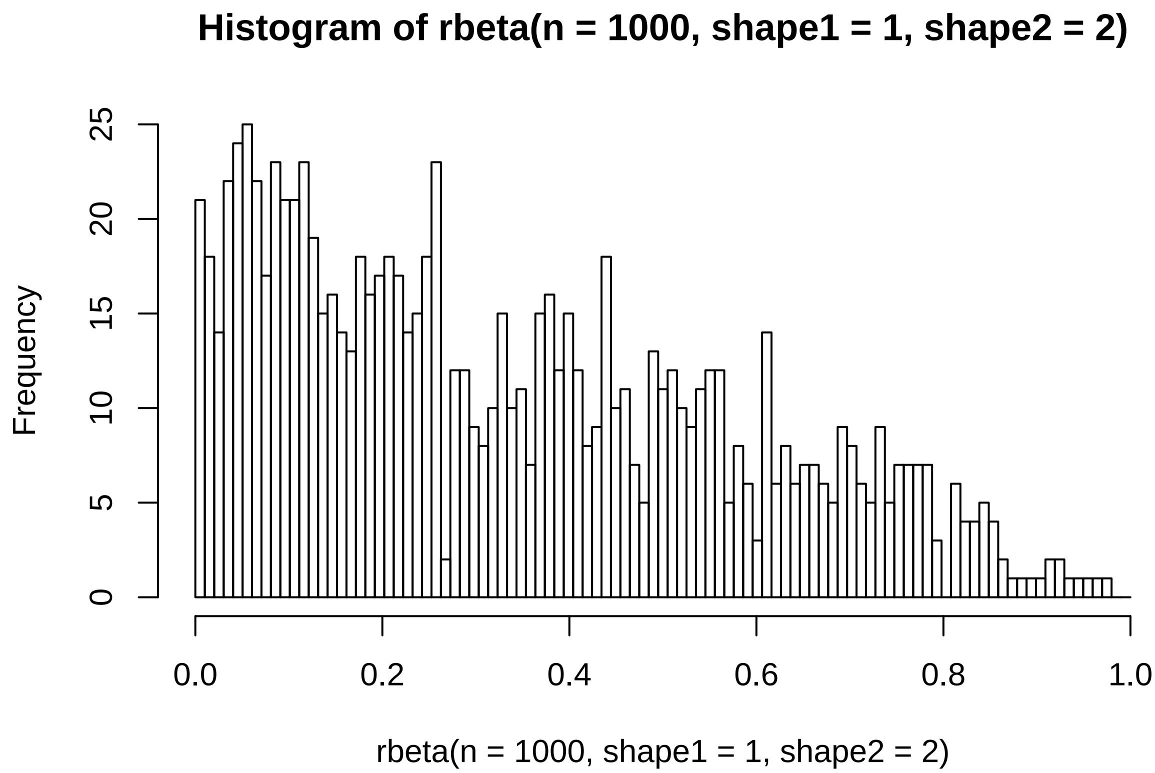
```r
# beta distribution
# bounded at 0 and 1
# analagous to a binomial, but result is a continuous distribution of probabilities
# parameter shape1 = number of successes + 1
# parameter shape2 = number of failures + 1
# interpret these in terms of a coin you are tossing

# shape1 = 1, shape2 = 1 = "no data"
hist(rbeta(n=1000,shape1=1,shape2=1),breaks=seq(0,1.0,length=100))
```



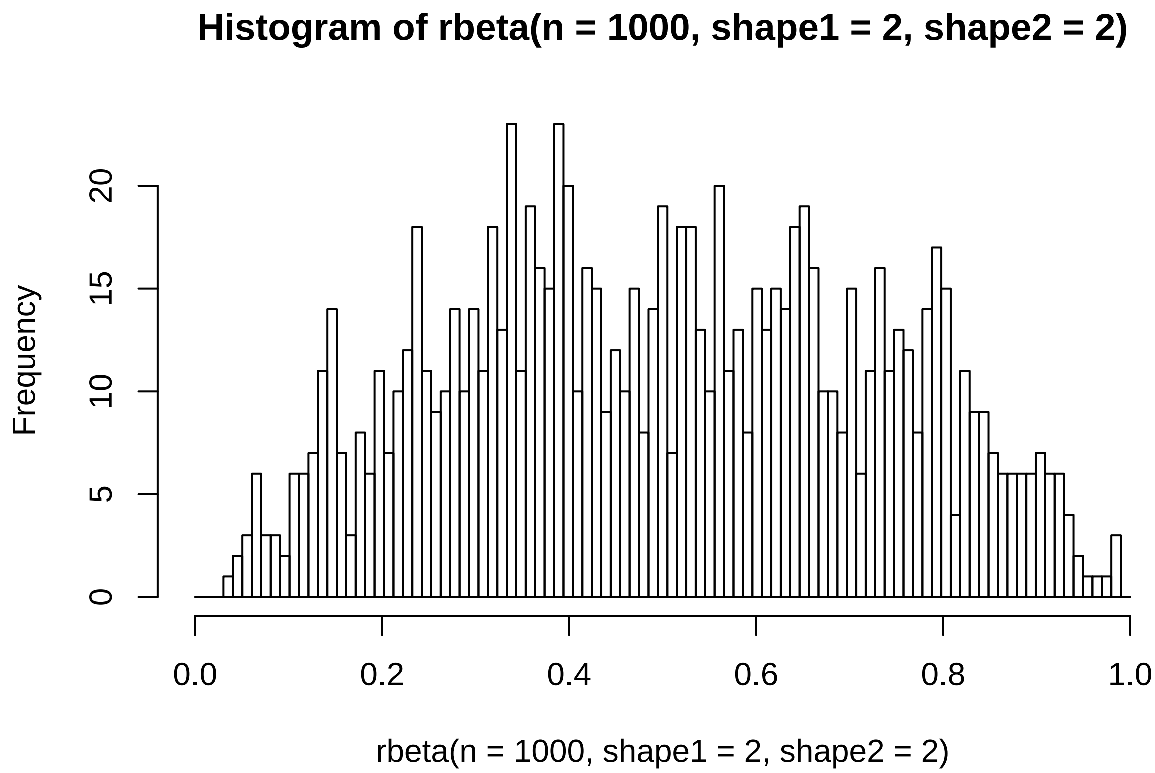

```r
# shape1 = 2, shape1 = 1 = "1 coin toss, comes up heads!"
hist(rbeta(n=1000,shape1=2,shape2=1),breaks=seq(0,1.0,length=100))
```



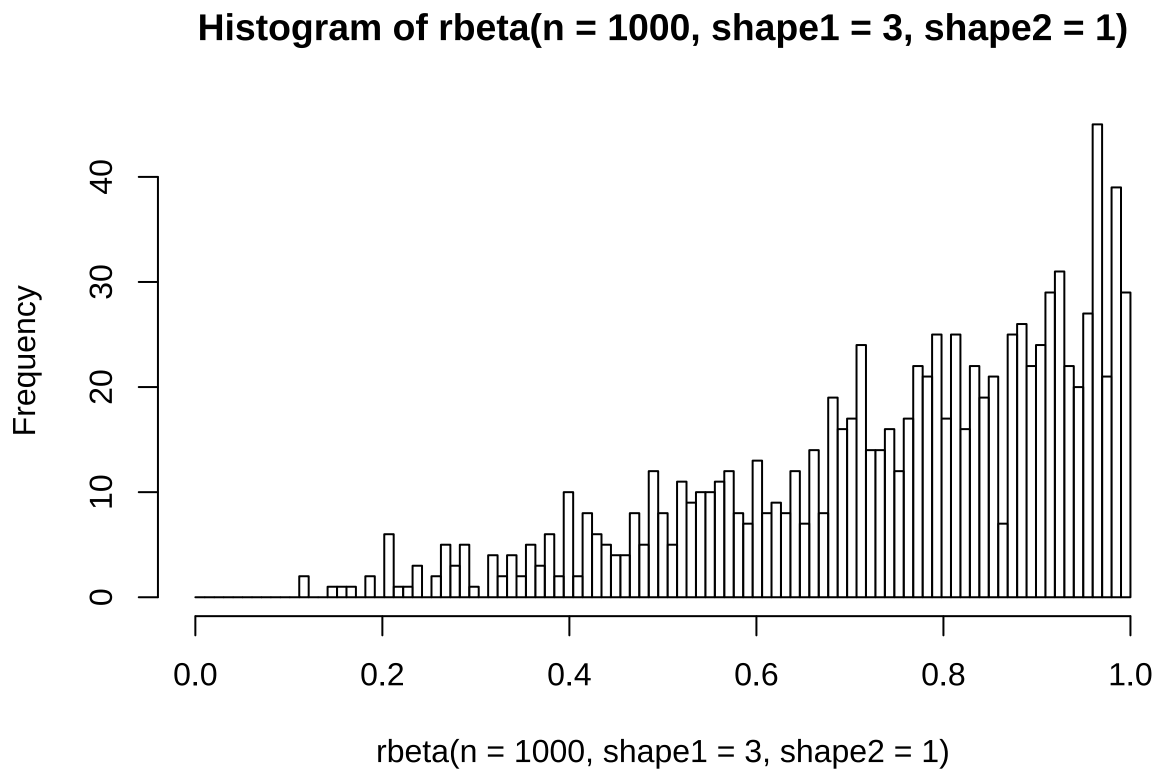

```
hist(rbeta(n=1000,shape1=1,shape2=2),breaks=seq(0,1.0,length=100))
```



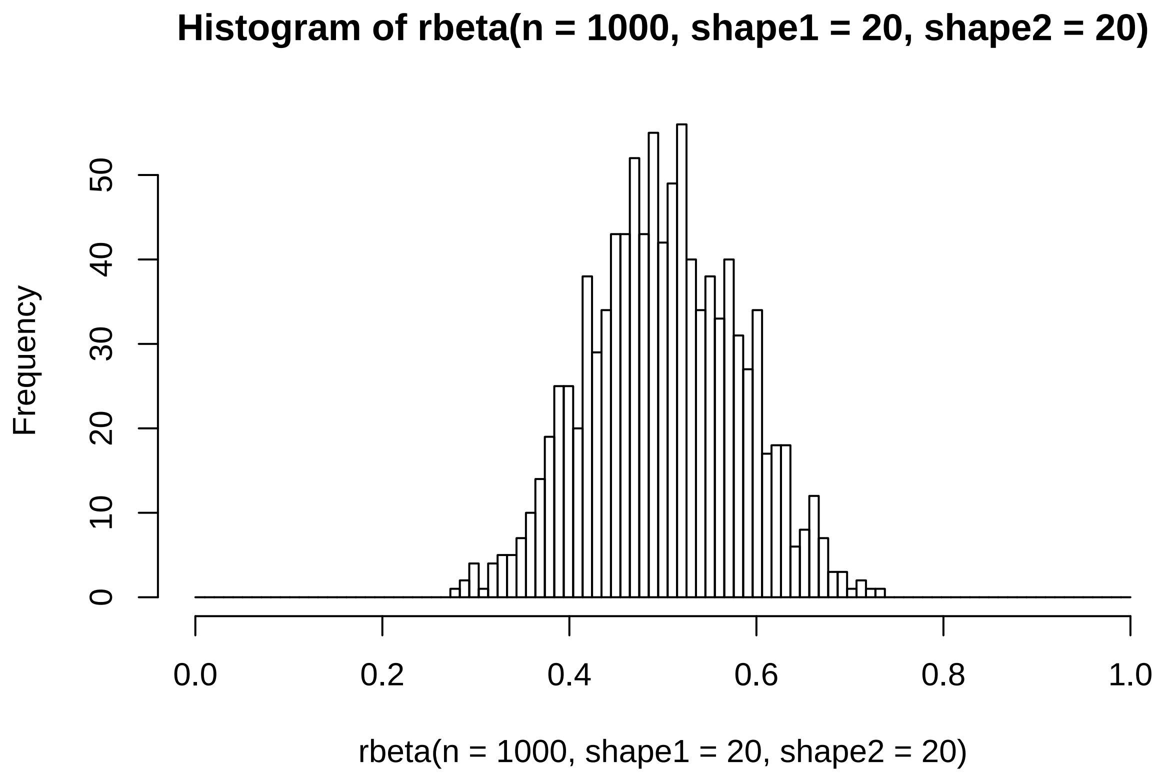

```
# two tosses, 1 head and 1 tail
hist(rbeta(n=1000,shape1=2,shape2=2),breaks=seq(0,1.0,length=100))
```



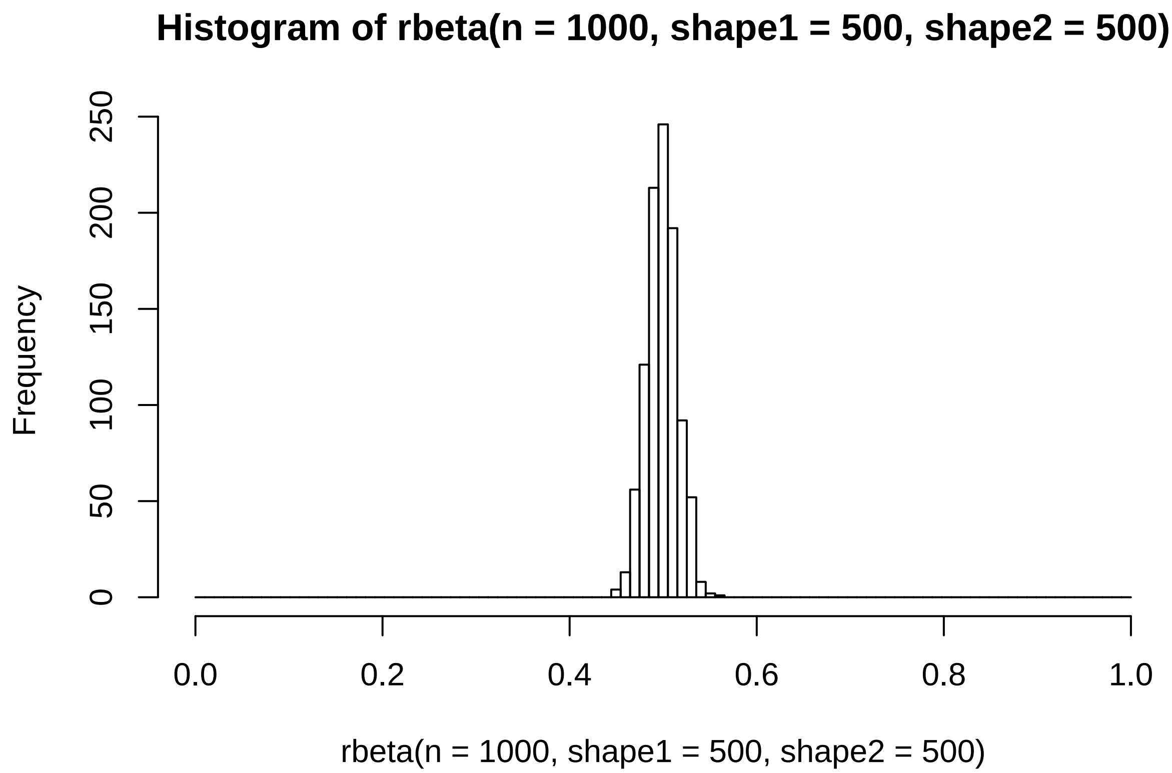

```
# two tosses, both heads
hist(rbeta(n=1000,shape1=3,shape2=1),breaks=seq(0,1.0,length=100))
```



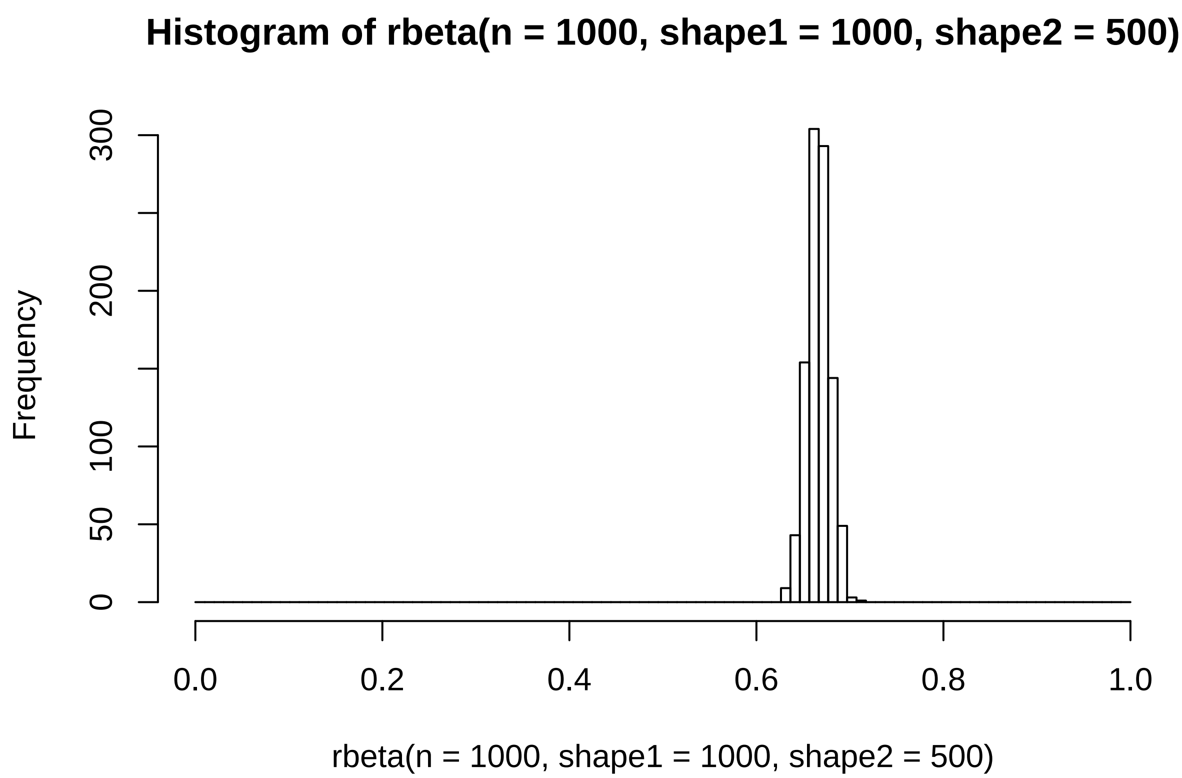

```
# let's get more data
hist(rbeta(n=1000,shape1=20,shape2=20),breaks=seq(0,1.0,length=100))
```



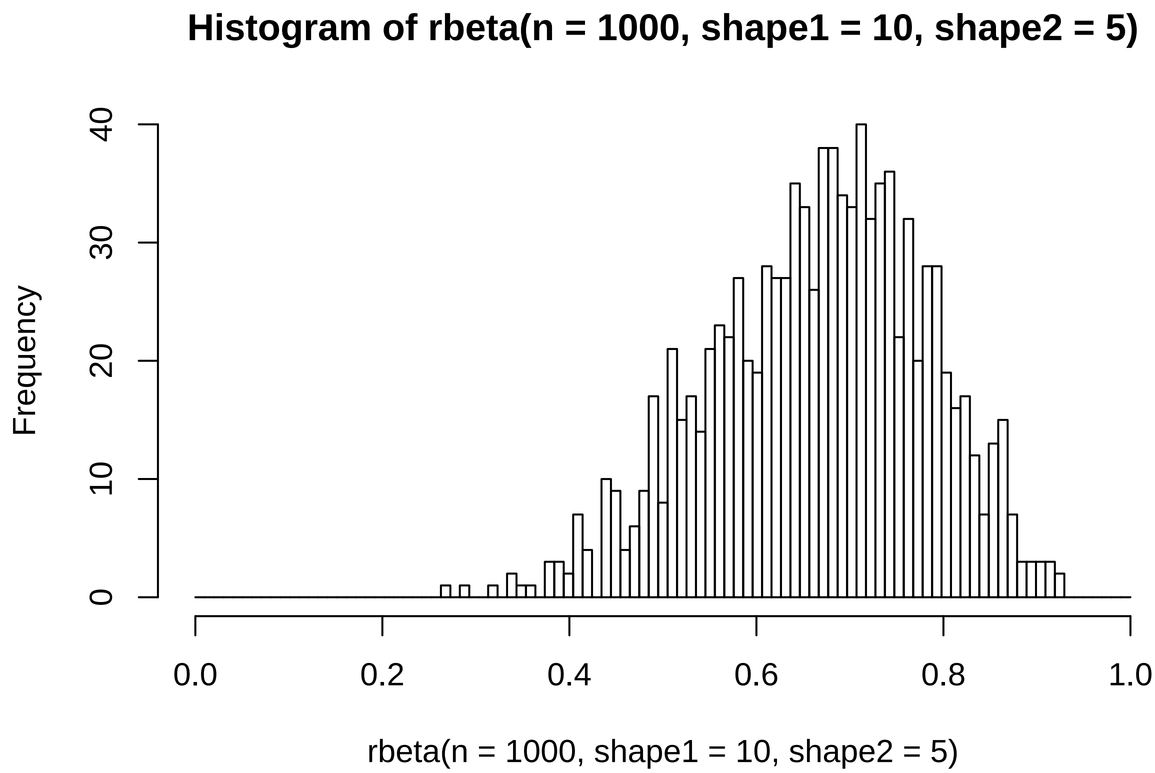

```
hist(rbeta(n=1000,shape1=500,shape2=500),breaks=seq(0,1.0,length=100))
```



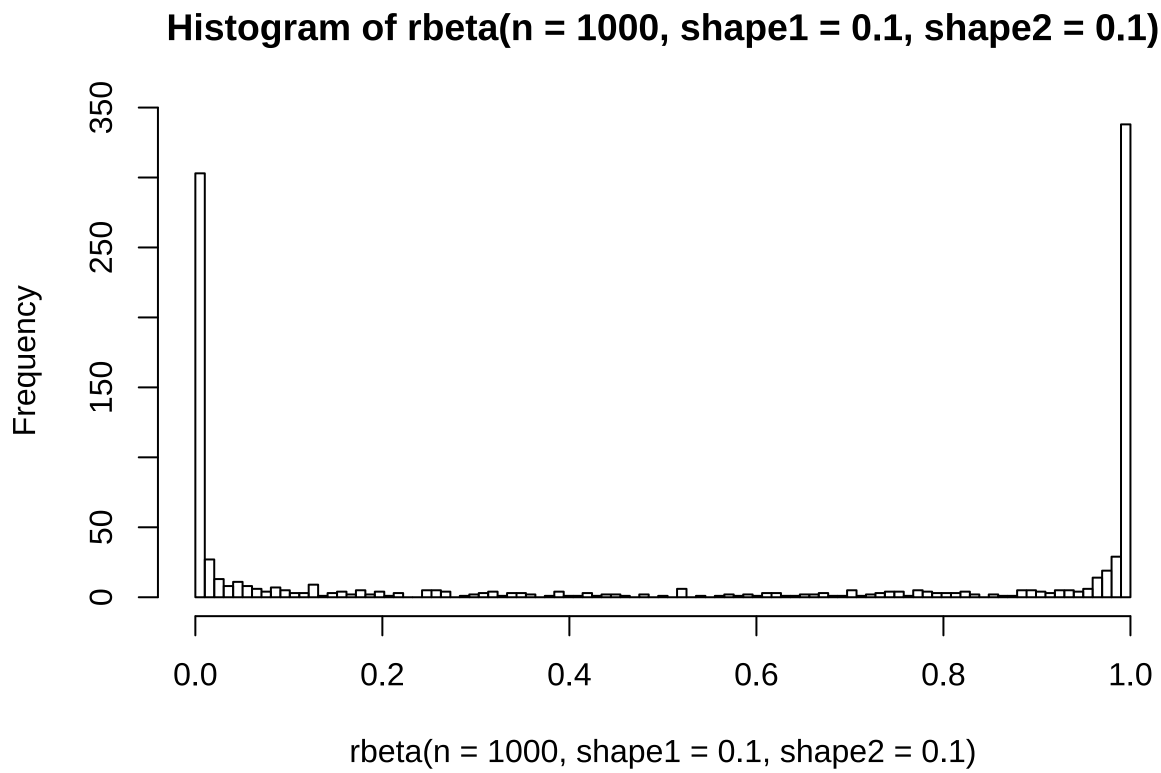

```r
# if the coin is biased
hist(rbeta(n=1000,shape1=1000,shape2=500),breaks=seq(0,1.0,length=100))
```



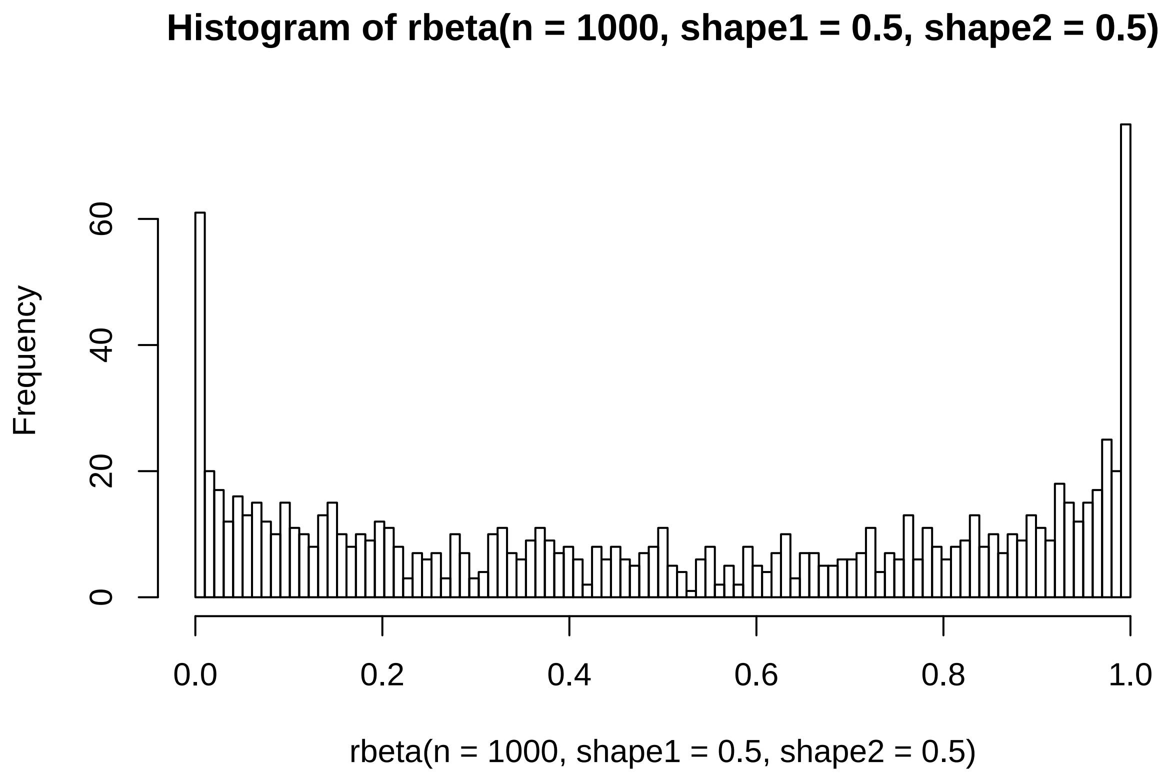

```
hist(rbeta(n=1000,shape1=10,shape2=5),breaks=seq(0,1.0,length=100))
```



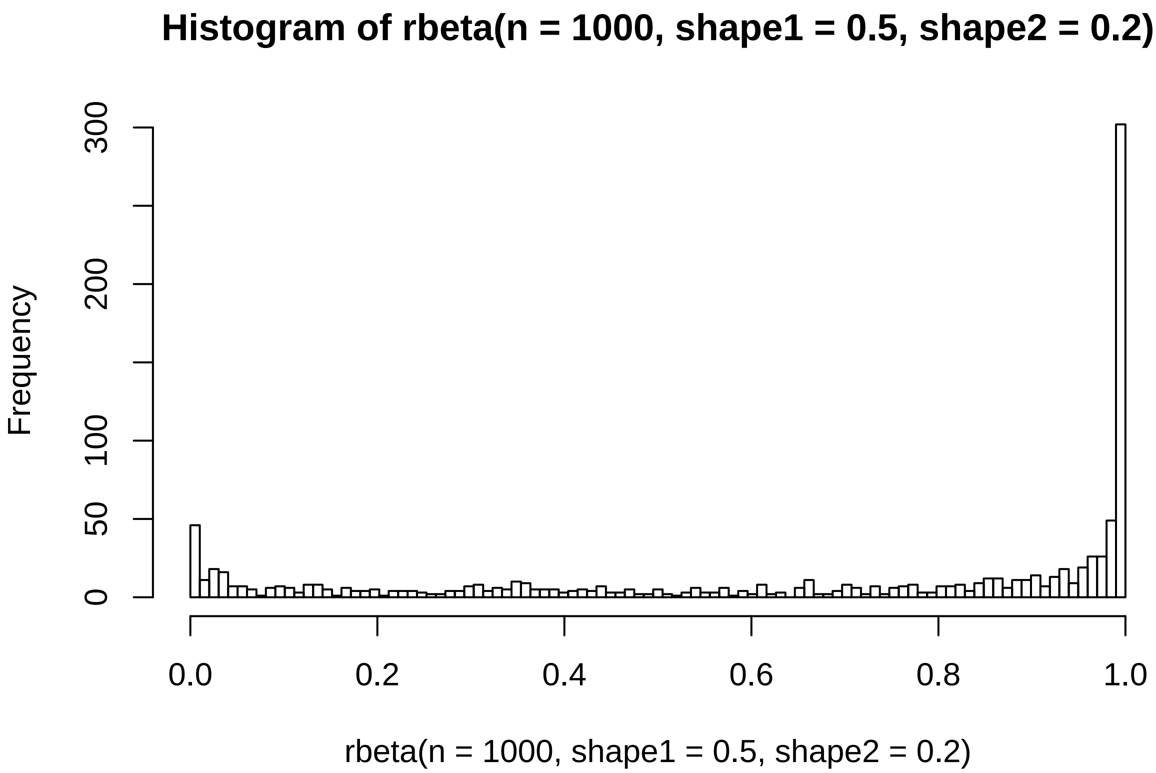

```
# shape parameters less than 1.0 give us a u-shaped distribution
hist(rbeta(n=1000,shape1=0.1,shape2=0.1),breaks=seq(0,1.0,length=100))
```



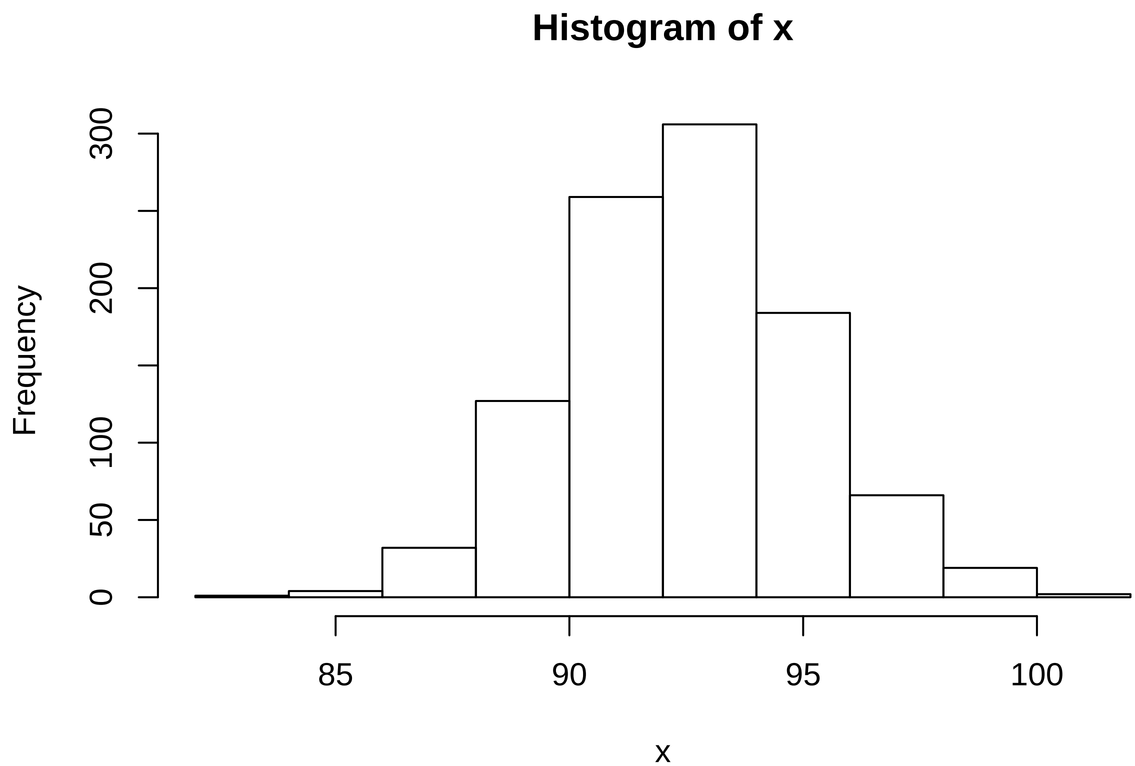

```
hist(rbeta(n=1000,shape1=0.5,shape2=0.5),breaks=seq(0,1.0,length=100))
```




```
hist(rbeta(n=1000,shape1=0.5,shape2=0.2),breaks=seq(0,1.0,length=100))
```




```
# estimating parameters from data
# maximum likelihood estimator theta versus P(data|theta)

# use fitdistr function, feeding it data and a distribution type)
library(MASS)
x <- rnorm(1000,mean=92.5,sd=2.5)
hist(x)
```




```
fitdistr(x,"normal")
```

```
##       mean            sd
##   92.50178159    2.55715063
##  ( 0.08086420) ( 0.05717963)
```

```
# compare to true parameters
# compare to parameters estimated from simple means and standard deviations
mean(x)
```

```
## [1] 92.50178
```

```
sd(x)
```

```
## [1] 2.55843
```

```
# but how do we "know" what distribution to fit?
fitdistr(x,"gamma")
```

```
##       shape           rate
##   1307.2331780     14.1319784
##  (   58.3900244) (    0.6313518)
```

```r
z <- fitdistr(x,"gamma")

# find components of z
str(z)
```

```
## List of 5
##  $ estimate: Named num [1:2] 1307.2 14.1
##   ..- attr(*, "names")= chr [1:2] "shape" "rate"
##  $ sd      : Named num [1:2] 58.39 0.631
##   ..- attr(*, "names")= chr [1:2] "shape" "rate"
##  $ vcov    : num [1:2, 1:2] 3409.395 36.858 36.858 0.399
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "shape" "rate"
##   .. ..$ : chr [1:2] "shape" "rate"
##  $ loglik  : num -2357
##  $ n       : int 1000
##  - attr(*, "class")= chr "fitdistr"
```

```r
# rate = 1/scale
# so here is the estimate of the mean
z$estimate[1]/z$estimate[2]
```

```
##    shape
## 92.50178
```

```r
# and here is the estimate of the variance
z$estimate[1]/z$estimate[2]^2
```

```
##    shape
## 6.545565
```